CS 189: Introduction to Machine Learning Spring 2014

Homework 4

He Ma SID: 22348372

Problem 1

1. As seen from textbook P.247,

$$\bigtriangledown_\beta - \sum_{i=1}^{n} [y_i \log \mu_i + (1 - y_i) \log(1 - \mu_i)] = X^T(\mu - y)$$

   So

$$g(\beta) = \bigtriangledown_\beta l = 2\lambda\beta + X^T(\mu - Y)$$

2. As seen from textbook p. 247,

$$H(\beta) = \frac{d}{d\beta} g(\beta)^T = 2\lambda I + X^T S X,\ S = diag(\mu_i(1 - \mu_i))$$

3. As seen from textbook p. 249,

$$\beta_{k+1} = \beta_k - H_k^{-1} g_k$$

4. (a) mu0 in code:

$$\mu_0 = \begin{bmatrix} 0.1192 \\ 0.2689 \\ 0.1192 \\ 0.2689 \end{bmatrix}$$

   (b) beta1 in code:

$$\beta_1 = \begin{bmatrix} -1.9240 \\ -1.3518 \\ 1.9662 \end{bmatrix}$$

   (c) mu1 in code:

$$\mu_1 = \begin{bmatrix} 0.9816 \\ 0.9323 \\ 0.5105 \\ 0.2125 \end{bmatrix}$$

   (d) beta2 in code:

$$\beta_2 = \begin{bmatrix} -2.0323 \\ -0.4154 \\ 1.1880 \end{bmatrix}$$

Problem 2

For this problem, I set the threshold for both gradient methods be norm($g(\beta)$) ¡ 1.

1. For all three ways of preprocessing data, I set the initial β 0.1 for all dimensions, $\lambda = 25$, $\eta = 10^{-5}$, and maximum iteration be 10000.

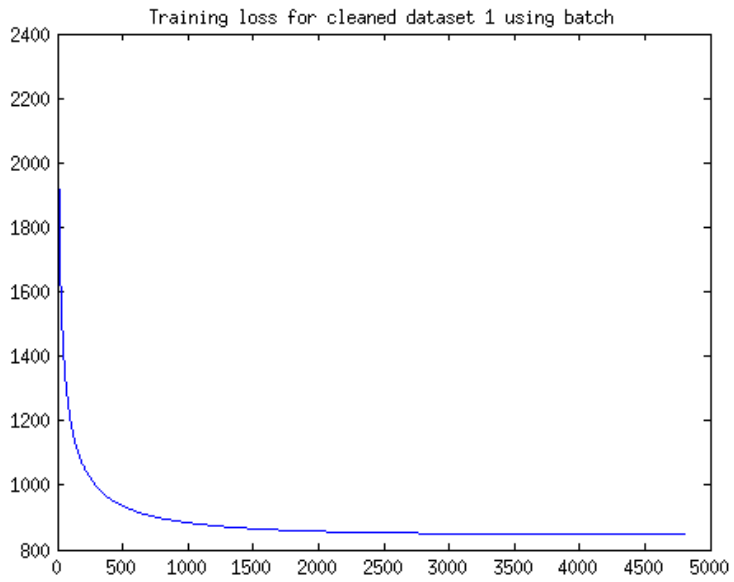

Figure 1: NLL for data set i)

Error rate for training set: 0.0855

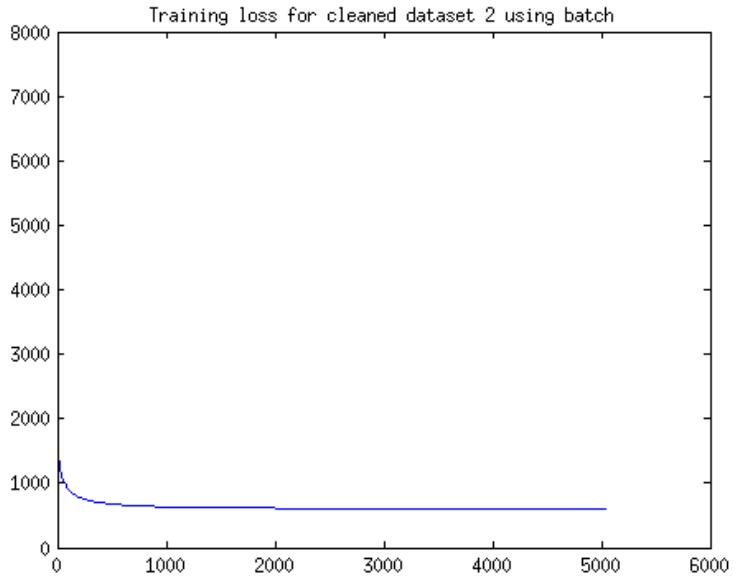

Figure 2: NLL for data set ii)

Error rate for training set: 0.0557

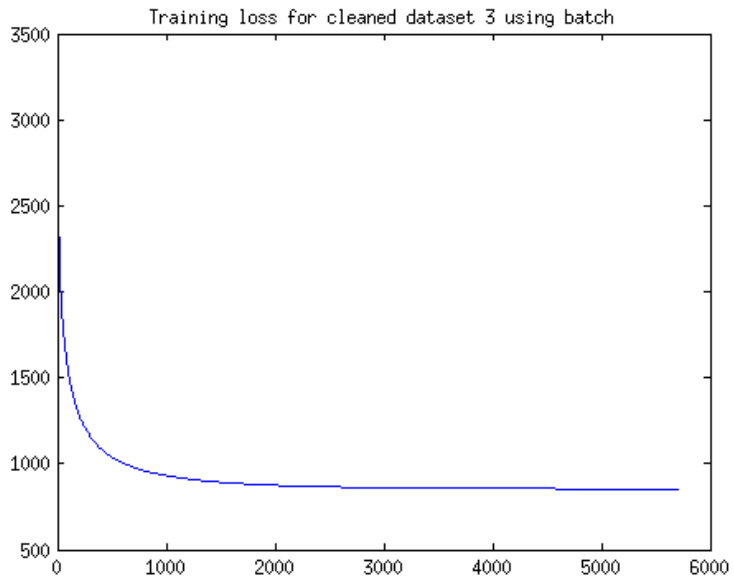

Figure 3: NLL for data set iii)

Error rate for training set: 0.0806

2. For method 1, I set the initial β 0.1 for all dimensions, $\lambda = 10$, $\eta = 10^{-4}$, and maximum loop be 30.

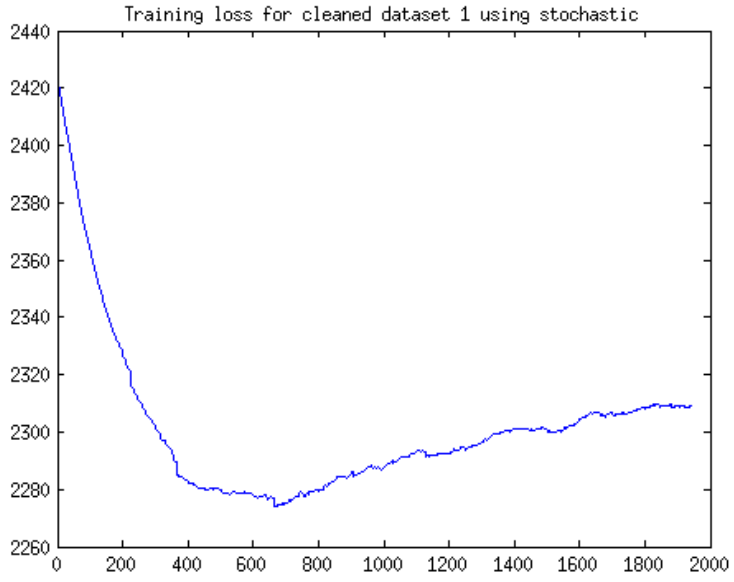

Figure 4: NLL for data set i)

Error rate for training set: 0.1009
For method 2, I set the initial β 0.1 for all dimensions, $\lambda = 10$, $\eta = 10^{-6}$, and maximum loop be 20.

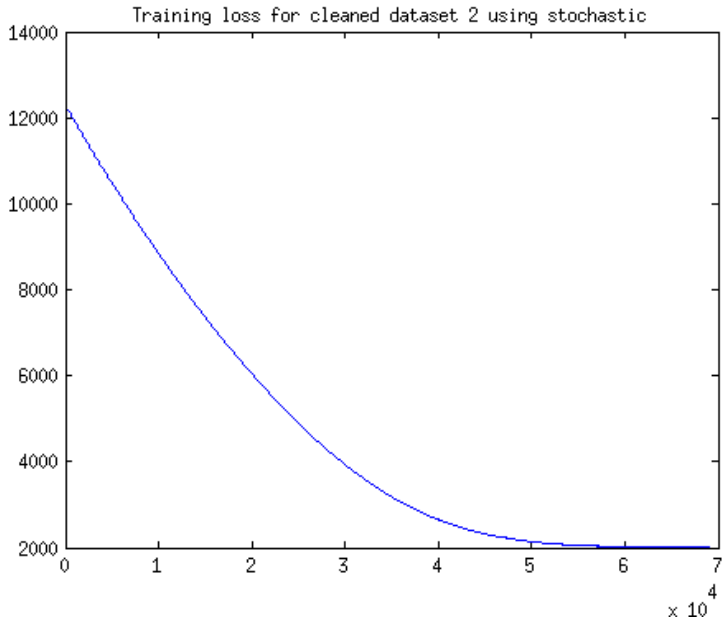

Figure 5: NLL for data set ii)

Error rate for training set: 0.3948
For method 3, I set the initial β 0.1 for all dimensions, $\lambda = 10$, $\eta = 10^{-4}$, and maximum loop be 30.

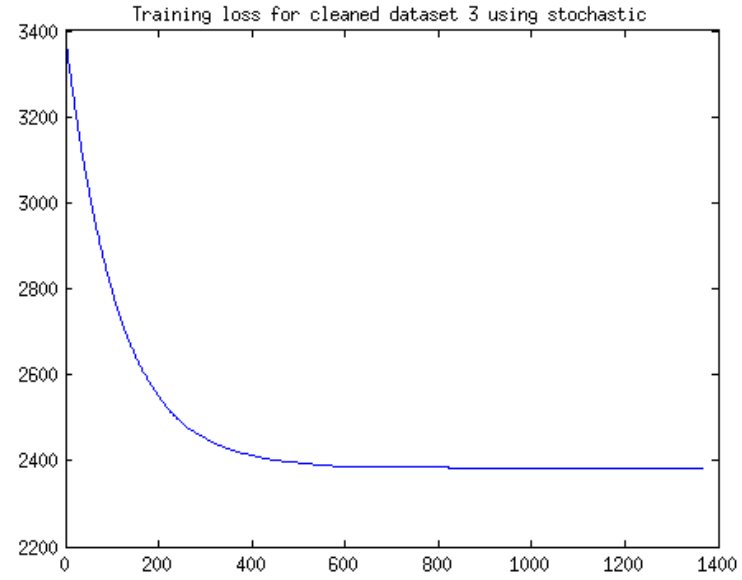

Figure 6: NLL for data set iii)

Error rate for training set: 0.1449

The curves for this method have more noise than the batch version. It is hard make NLL converge. The value jumps around along the local minimum's direction.

3. For all three ways of preprocessing data, I set the initial β 0.1 for all dimensions, $\lambda = 10$, $\eta = 10^{-5}$, and maximum iteration be 30.

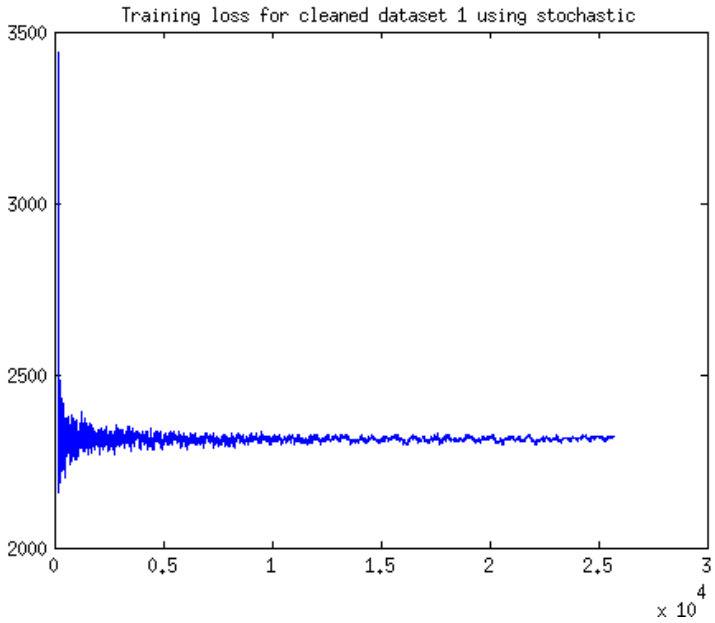

Figure 7: NLL for data set i)

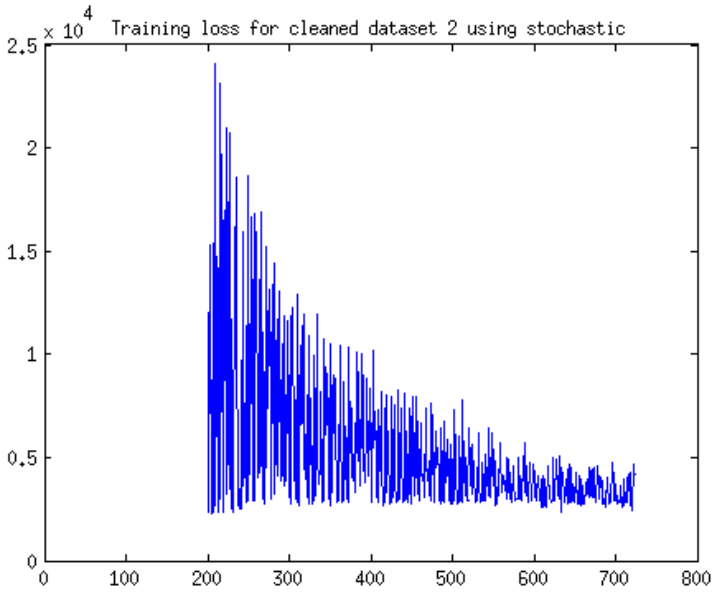

Figure 8: NLL for data set ii)

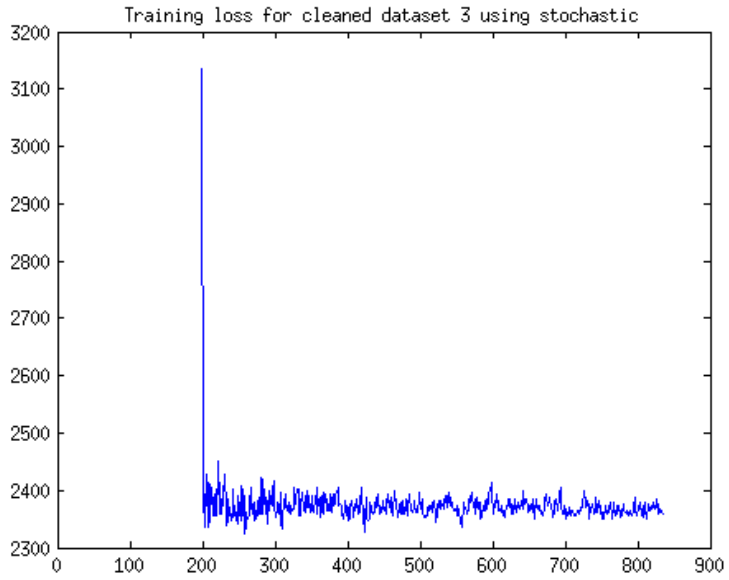

Figure 9: NLL for data set iii)

This method helps the curve to converge. However, there seems to be more noise. Also since the method takes about 10000 iterations on average, the weigth $\frac{1}{t}$ might be too small for the later iterations.
Some other strategies I can think of are:
1. Scale $\frac{1}{t}$ for the t^{th} loop of the whole data set rather than the t^{th} data point. So it will take more weight for the later iterations.
2. For each iteration of stocahstic gradient decent, use k data points rather than one, so the direction will be more accurate.

4. I used 10 folds cross validation to determine the best λ, η pair for each of the model. And data set 2 has lowest validation error, which is 0.0562. It is achieved by $\lambda = 0.96$, and $\eta = 10^{-5}$. The error rate for the other two data set are around 0.08. So the 2nd preprocessing method is better.
The test set error rate is 0.05963 computed by Kaggle.